



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

출원 번호 : 10-2003-0012902  
Application Number

출원 년 월 일 : 2003년 02월 28일  
Date of Application FEB 28, 2003

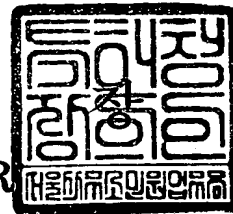
출원인 : 삼성전자주식회사 외 1명  
Applicant(s) SAMSUNG ELECTRONICS CO., LTD., et al.



2003 년 08 월 22 일

특 허 청

COMMISSIONER



## 【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【제출일자】	2003.02.28
【발명의 명칭】	필드레벨 트리를 이용한 패킷분류장치 및 방법
【발명의 영문명칭】	Apparatus and method for packet classification using Field Level Trie
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【출원인】	
【명칭】	폴리테크닉 대학교
【출원인코드】	5-1998-082774-1
【대리인】	
【성명】	정홍식
【대리인코드】	9-1998-000543-3
【포괄위임등록번호】	2003-002208-1
【발명자】	
【성명의 국문표기】	장관송
【성명의 영문표기】	ZHANG, Guansong
【주소】	미국 , 뉴욕 11201, 브룩클린, 6 메트로테크 센터, 폴리테크닉 대학교 전기 및 컴퓨터 공학과
【주소의 영문표기】	Department of Electrical and Computer Engineering Polytechnic University 6 Metrotech Center, Brooklyn, New York 11201, U.S.A.
【국적】	CN
【발명자】	
【성명의 국문표기】	차오 조나단
【성명의 영문표기】	CHAO, H. Jonathan
【주소】	미국 , 뉴욕 11201, 브룩클린, 6 메트로테크 센터, 폴리테크닉 대학교 전기 및 컴퓨터 공학과

【주소의 영문표기】	Department of Electrical and Computer Engineering Polytechnic University 6 Metrotech Center, Brooklyn, New York 11201, U.S.A.	
【국적】	US	
【발명자】		
【성명의 국문표기】	정진우	
【성명의 영문표기】	JOUNG, JIN WOO	
【주민등록번호】	700129-1702715	
【우편번호】	449-907	
【주소】	경기도 용인시 기흥읍 신갈리 도현마을 현대아파트 201동 501호	
【국적】	KR	
【발명자】		
【성명의 국문표기】	박우중	
【성명의 영문표기】	PARK, WOO JONG	
【주민등록번호】	710826-1074410	
【우편번호】	156-772	
【주소】	서울특별시 동작구 사당2동 극동아파트 112-1513	
【국적】	KR	
【심사청구】	청구	
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정 에 의한 출원심사를 청구합니다. 대리인 정홍식 (인)	
【수수료】		
【기본출원료】	20 면	29,000 원
【가산출원료】	7 면	7,000 원
【우선권주장료】	0 건	0 원
【심사청구료】	13 항	525,000 원
【합계】	561,000 원	
【첨부서류】	1. 요약서·명세서(도면)_1통	

**【요약서】****【요약】**

필드레벨 트리를 이용한 패킷분류장치 및 방법이 개시된다. 본 패킷분류장치는 분류를 위해 멀티필드를 갖는 패킷을 계층적 구조로 필드별로 조직화한 필드레벨트리를 생성하여 유지하는 주처리부, 및 쿼리와 갱신업무를 수행하며, IP 소스/목적지 어드레스 룩업으로 대표되는 프리픽스 룩업을 처리하는 1차 분류부와 상기 1차 분류부의 결과에 의거하여 그에 속한 레인지 룩업범위를 처리하기 위해 해당 필드별로 분류를 진행하는 2차 분류부를 구비하는 분류엔진을 구비한다. 본 발명에 의하면, 필드단위의 트리를 전개함으로써 뛰어난 쿼리 성능이 확보된 초고속 네트워킹을 위한 패킷 분류를 구현할 수 있다. 또한, 약 50만개의 분류기 룰을 처리할 수 있다.

**【대표도】**

도 5

**【색인어】**

필드레벨트리, 패킷분류

**【명세서】****【발명의 명칭】**

필드레벨 트리를 이용한 패킷분류장치 및 방법{Apparatus and method for packet classification using Field Level Trie}

**【도면의 간단한 설명】**

도 1은 종래의 패킷분류방법에서 그리드-오브-트리의 데이터 구조도를 나타내는 도면,

도 2는 종래의 패킷분류방법에서 레인지 룩업의 원리를 나타내는 도면,

도 3은 종래의 패킷분류방법에서 리커시브 플로우(recursive flow) 분류의 원리를 나타내는 도면,

도 4는 본 발명에 따른 필드레벨 트리를 이용한 패킷분류장치의 구성도,

도 5는 도 4에서 분류엔진의 구성도,

도 6은 표 2의 분류기를 필드레벨트리로 구성한 데이터 구조도,

도 7은 도 6의 프리픽스 룩업을 위한 데이터구조를 도 5의 분류엔진내에 TCAM(Ternary Content Addressable Memory)을 사용해서 단축한 구성도,

도 8a는 도 5의 분류엔진내의 레인지룩업을 위해 마이크로 엔진에서 수행되는 k-way search를 설명하기 위한 원리도, 그리고

도 8b는 도 5의 분류엔진내의 레인지룩업을 위한 마이크로 엔진에서 수행되는 k-way search를 설명하기 위한 원리도이다.

**【발명의 상세한 설명】****【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

- <10> 본 발명은 통신시스템에서 패킷 등의 항목을 처리하는 분야에 관한 것으로, 라우터에서 패킷을 분류하는데 있어 필드레벨 트리를 이용하여 패킷을 분류하는 패킷 분류방법 및 그 방법을 실현시키기 위한 패킷분류장치에 관한 것이다.
- <11> 인터넷과 같은 통신시스템에서 패킷을 처리함에 있어서, 통상은 목적지 어드레스를 계산하여 패킷경로상의 중간 노드 각각에서 어느 출력 노드 또는 링크상으로 패킷을 포워드해야 하는지를 결정하게 된다. 각종의 통신 시스템에서 각 패킷의 헤더에 있는 목적지 어드레스, 발신지 어드레스 또는 기타 데이터에 따라 다양한 형태의 서비스가 제공된다. 서비스 형태의 차이라는 것은 패킷이 처리 또는 포워드되는 우선순위나 그 전송에 대해 지불해야할 요금 또는 어떤 발신지에 대해서는 패킷처리를 거부하는 것 등이 있다.
- <12> 오늘날의 시스템에서는 엄청난 양의 패킷(일반적으로는 데이터 항목)을 처리해야 되기때문에, 패킷과 함께 수신된 기준을 알아 내고 그에 따라 어떤 형태의 처리를 선택해야 하는지를 결정하는 일은 아주 짧은 시간내에 고속으로 이루어져야만 한다.
- <13> 미래의 IP 네트워크상에서 사용자에게 서비스 레벨 어그리먼트(service level aggrements), VPN, QoS 등의 더욱 발전된 서비스를 제공하기 위해서 라우터 등에서 입력되는 IP 패킷을 원하는 규칙에 따라서 분류할 수 있어야 한다. 이를 패킷 분류(packet

classification)라고 하며, 패킷분류는 결국 패킷내 여러 필드의 값을 참조해야 하는 다중 필드 룩업(multi-field lookup)이다.

<14> 다시 말하면, 패킷분류는 기존의 IP 목적지 어드레스 룩업(IP destination address lookup)과는 달리 하나의 패킷내 소스 어드레스(source address), 목적지 어드레스(destination address), 프로토콜 ID(protocol ID), 포트번호(port number) 등의 여러 필드를 참조해서 패킷을 처리해야 한다. 그러므로, 기본적으로 더 많은 시간과 메모리를 필요로 하는 것은 물론이고 이를 해결할 방법에 대한 연구도 부족한 실정이다.

<15> 도 1 내지 도 3은 종래의 패킷분류방법의 일예를 나타내는 도면이다.

<16> 기존 방법 구성을 위한 분류기의 예를 표1에 나타낸다.

<17> 【표 1】

Filter	F1	F2
R <sub>1</sub>	00*	11*
R <sub>2</sub>	00*	1*
R <sub>3</sub>	10*	1*
R <sub>4</sub>	0*	01*
R <sub>5</sub>	0*	10*
R <sub>6</sub>	0*	1*
R <sub>7</sub>	*	00*

<18> 종래의 패킷분류방법으로는 도 1에 나타낸 것같은 그리드-오브-트리(grid-of-trie)로 대표되는 전통적인 방법과, 도 2의 레인지 룩업(range lookup)으로 대표되는 기하학적인 방법, 그리고 도 3의 리커시브 플로우 분류(recursive flow classification)로 대표되는 경험적인 방법등이 있다.

<19> 이 중, 그리드-오브-트리 데이터구조는 표준 계층 트리와 셋-프루닝 트리의 장점을 결합한 것이고,  $O(dW)$ 의 쿼리 시간 복잡도와  $O(NdW)$ 의 저장 복잡도를 갖는다. 이 특징은 스위칭 포인터를 데이터구조에 도입함으로써 얻어진다. 그러나, 스위칭 포인터는 비트레벨에서 조합되기 때문에,  $O(NdW)$ 의 저장 복잡도를 보증하고, 쿼리 과정은 비트씩 행해져야 한다. 실행이 요구되는 실제의 애플리케이션에서 비트씩의 쿼리 과정(또는 분류)은 받아 들여지지 않는다.

<20> 패킷을 분류하는데 있어서 분류속도, 메모리크기, 분류규칙수, 참조 필드수, 규칙 갱신시간 및 최악의 조건시 성능이 주요 항목으로 고려된다.

<21> 즉, 패킷분류는 사용가능한 메모리를 사용하여 최대의 성능을 확보하는 것으로서, 성능(High performance)과 비용(storage limitation)의 문제를 해결하는 것이 중요하다.  
**【발명이 이루고자 하는 기술적 과제】**

<22> 따라서, 본 발명은 상기 문제점을 해결하기 위해, 종래에 사용된 비트단위의 트리가 아닌 필드단위의 트리를 전개하고, TCAM 과 k-way 검색을 이용하여 각각 프리픽스 구체화와 레인지 구체화를 다루는, 쿼리 성능을 개선한 필드레벨 트리(FLT: field-level trie)를 이용한 패킷분류장치 및 방법을 제공하는 것을 목적으로 한다.

#### **【발명의 구성 및 작용】**

<23> 상기 목적을 달성하기 위해, 분류를 위해 멀티필드를 갖는 패킷을 계층적 구조로 필드별로 조직화한 필드레벨트리를 생성하여 유지하는 주처리부; 및 쿼리와 갱신업무를 수행하며, IP 소스/목적지 어드레스 룩업으로 대표되는 프리픽스 룩업을 처리하는 1차 분류부와 상기 1차 분류부의 결과에 의거하여 그에 속한 레인지 룩업범위를 처리하기 위



해 해당 필드별로 분류를 진행하는 2차 분류부를 구비하는 분류엔진을 구비하는 것을 특징으로 하는 필드레벨트리를 이용한 패킷분류장치가 제공된다.

- <24>      상기 분류엔진은 분류 프로세서와 메모리를 구비한다.
- <25>      또한, 상기 주처리부와 상기 분류엔진은 브로드캐스팅 버스로 연결되어 있다.
- <26>      상기 1차 분류부는 프리픽스 형태의 필드를 저장하고 그것을 검색하는 TCAM을 사용하는 것이 바람직하다.
- <27>      그리고, 상기 2차 분류부는 용도 및 스펙에 따라 적당한 k값을 갖는 k-way search 방법을 사용하는 것이 바람직하다.
- <28>      또한, 상기 주처리부는 상기 브로드 캐스팅 버스를 통해 상기 복수의 분류엔진에 갱신지시를 보내고, 상기 갱신지시를 받은 분류엔진은 상기 메모리의 콘텐츠를 변경하도록 지시한다.
- <29>      상기 필드레벨트리는 제 1그룹의 필드가 상위레벨에 나타나고 제 2그룹의 필드는 하위레벨에 나타나는 구조로 조직화된다.
- <30>      상기 필드레벨트리는 어느 레벨에서 2개의 노드가 공통의 자식노드를 가지면, 단지의 한개의 노드가 생성되고 그것을 공유한다.
- <31>      상기 필드레벨트리는 프리픽스 특업을 위한 레벨은 서로 결합된 프리픽스쌍을 갖는 단지 하나의 레벨로서 존재한다.
- <32>      상기 다른 목적을 달성하기 위해, 라우팅 시스템의 패킷 분류방법에 있어서,

주어진 멀티필드를 갖는 패킷을 필드별로 전개하여 필드레벨트리를 형성하는 제 1단계;  
 상기 필드레벨트리를 이용하여 패킷 분류 규칙들에 대해 프리픽스 룩업을 행하는 제 2단  
 계; 및 프리픽스 룩업이 행해진 이후, 레인지 룩업을 행하는 제 3단계;를 포함하는 필드  
 레벨트리를 이용한 패킷 분류방법이 제공된다.

<33>       다음은 본 발명에 따른 본 발명에 따른 필드레벨 트리를 이용한 패킷분류장치에 대  
 하여 설명한다.

<34>       도 4는 본 발명에 따른 필드레벨 트리를 이용한 패킷분류장치의 구성도이다.

<35>       도 4에 나타난 것같이, 본 발명에 따른 필드레벨 트리를 이용한 패킷분류장치는 주  
 처리부(10)와 복수의 개별 분류엔진(20)을 구비한다.

<36>       주처리부(10)는 FLT(Field Level Trie) 데이터 구조를 구성 및 관리하고, 이 정보  
 를 가공하여 각각의 분류엔진(20)에 제공한다. 각각의 인터페이스에 속한 분류엔진(20)  
 은 입력되는 개별 패킷을 실제로 분류한다.

<37>       고속/코어 라우터에서, 패킷 분류는 각각의 라인카드에서 일반적으로 병렬로 행해  
 진다. 그러므로, 각각의 라인카드는 적어도 하나의 내장된 분류엔진을 갖는다.   전송  
 링크의 요구 대역폭으로 인해, 이들 분류엔진은 쿼리 기능에 최적화된다.

<38>       본 발명에 따른 필드레벨 트리를 이용한 패킷 분류방식은 TCAM 기술과 k-way 검색  
 을 결합하여 프리픽스 구체화와 레인지 구체화를 다루는 하이브리드 방식이다.

<39>       그러나, 분류는 동적이고, 데이터 구조는 때때로 변경되어야 할 필요가 있다. 분류  
 엔진에서 데이터 조직화는 룰의 삽입 또는 삭제와 같은 업데이트에 적당하지 않을 수 있  
 다.

- <40> 분류는 각각의 라인 카드에서 동일하고, 중심이 되는 곳에서 실행된 업데이트 기능을 갖는 것이 더 바람직하다.
- <41> 주처리부(10)는 통상 강력한 프로세서를 구비하고, 라인카드와는 별개의 보드에 존재한다.
- <42> 주처리부(10)는 분류를 위해 사용된 전체 데이터 구조를 유지하고, 분류가 업데이트될 필요가 있을 때 변경한다. 도 6 내지 도 8은 본 발명에 따른 데이터 구조를 나타내며, 도 6에 나타낸 데이터 구조는 주처리부(10)에서 유지되고, 도 7, 도 8a 및 b에 나타낸 데이터 구조는 분류엔진(20)에 의해 사용된다.
- <43> 도시된 데이터 구조에서, 각각의 노드에 룰을 저장할 필요는 없다. 도 8a 및 b에 나타낸 것같이, 프리픽스쌍, k-way 검색값 및 포인터가 분류 엔진(20)에서 유지된다.
- <44> 주처리부(10)는 갱신 지시를 브로드 캐스팅 버스(12)를 통해 모든 분류엔진(20)에 보낸다. 이들 갱신 지시는 분류엔진(20)이 메모리(및/또는 TCAM)의 콘텐츠를 변경하도록 지시하기 위해 사용된다.
- <45> 도 5는 도 4에서 분류엔진(20)의 구성도이다.
- <46> 개별 분류엔진(20)은 IP 소스/목적지 어드레스 룩업으로 대표되는 프리픽스 룩업을 처리하는 부분(21)과 1차 분류된 결과에 의거하여 그에 속한 2차 분류를 처리하는 부분(22)으로 구성된다. 2차분류에 있어서, 레인지 룩업범위를 처리하기 위해 해당 필드별로 분류를 진행하기 위한 하위 마이크로 엔진이 구비된다.
- <47> 2차 분류를 위해서는 용도 및 스펙에 따라 적당한 k값을 갖는 k-way search방법을 사용한다.

- <48> 분류엔진(20)은 분류 프로세서(32)와 TCAM(41)과 일반적인 메모리(SDRAM 및/또는 SSRAM)(42)를 구비하는 외부 메모리로 구성된다.
- <49> 분류 프로세서(32)는 그 내부에 마이크로 엔진(31)과 메모리 인터페이스(30)의 주요 구성요소를 구비하고 있다. 마이크로 엔진(31)은 특정 애플리케이션부 또는 일반적인 목적의 RISC 프로세서이다. 데이터 구조는 외부메모리(40)에 저장되어 있다. 내부 마이크로 엔진(31)은 쿼리 동작에 대하여 제어 기능을 행하고 메모리 인터페이스(32)를 통하여 데이터에 접속한다.
- <50> 추출된 모든 필드를 갖는 패킷 헤더는 좌측에서 분류 프로세서(30)로 들어간다. 첫번째 마이크로 엔진(31)은 프로픽스 포맷으로 구체화된 첫번째 k필드를 취하여, TCAM(41)에 보낸다. TCAM(41)은 검색을 실행하고 그 결과를 첫번째 마이크로 엔진(31)에 되돌려 보낸다. 그 결과를 몇개의 마이크로 엔진이 레인지에서 구체화된 다음의 필드의 k-way 검색에 대하여 존재하는 두번째 단에 보낸다. 도 5에서 하나의 마이크로 엔진은 두번째 단에서 각각의 필드에 전용된다. 그러나, 실제의 동작에서, 하나의 마이크로 엔진은 몇개의 필드를 수용할 수 있다. 마이크로엔진의 수는 각각의 필드에서 검색을 위해 필요한 외부 메모리 대역폭, 마이크로엔진의 속도 및 평균대역폭에 의해 결정된다.
- <51> 쿼리기능이외에도, 분류엔진은 주처리부에서 송신된 갱신지시에 의거하여 그 외부 메모리의 콘텐츠를 변경할 필요가 있다. 이들 2개의 업무사이에 컨텐션을 해결하는 2개의 방법이 있다. a) 분류엔진은 쿼리 또는 갱신을 언제든지 실행할 수 있는 태스크를 인터리빙한다. 시간소비에 관하여 갱신비용이 적을 때 실현가능하다. b) 메모리에서 데이터 구조의 2개의 복사본을 만든다. 하나는 쿼리를 위해 사용되고, 다른 것은 갱신을

위해 사용된다. 하나의 갱신동작이 종료되면, 이 카피는 쿼리용으로 전환되고, 다른 것은 갱신을 위해 사용된다. 이 방법은 갱신비용이 인터리빙 방법의 쿼리 성능을 매우 낮게 할 만큼 클때 적용한다.

<52> 다음은 본 발명에 따른 필드레벨트리 분류구조의 실시예와 이에 따른 필드레벨트리 분류방법에 대하여 도면을 참조하여 상세히 설명한다.

<53> 필드 레벨 트리 분류 구조는 다중 필드를 갖는 분류기의 분류를 목표로 하고 각각의 필드는 개개의 프리픽스 구체화 또는 레인지 구체화에서 구체화된다.

<54> 4개의 필드와 7개의 규칙을 갖는 분류기의 일예가 표 2에 표시되어 있다.

<55> 【표 2】

Rule	F1	F2	F3	F4
R <sub>1</sub>	00*	110*	6	[10,12]
R <sub>2</sub>	00*	11*	[4,8]	15
R <sub>3</sub>	10*	1*	7	9
R <sub>4</sub>	0*	01*	10	[10,12]
R <sub>5</sub>	0*	10*	[4,8]	15
R <sub>6</sub>	0*	1*	10	[10,15]
R <sub>7</sub>	*	00*	7	15

<56> 표 2에서, 처음 2개의 필드(F1, F2)는 프리픽스로 구체화되고 마지막 2개의 필드(F3, F4)는 레인지로 구체화된다. 이를 FLT로 구성한 데이터 구조도가 도 6에 도시되어 있다.

<57> 각 개별 패킷은 각 레벨에서 해당하는 필드에 정한 규칙을 따라서 분류된다. 주어진 모든 레벨의 필드를 거치면 분류작업이 끝나 개별패킷에 정해진 규칙을 부여하게 된다.

- <58> 다음은 도 6의 필드레벨트리의 데이터 구조도를 상세히 설명한다. 필드레벨트리의 데이터 구조는 다음의 속성을 갖도록 정의된다.
- <59> 1. 필드레벨트리는 계층적 구조로 필드씩 조직화된다. 트리의 깊이는 필드의 수 (d)와 같다. 도 6의 일예에서 F1에서 F4까지 조직화된 4개의 노드 레벨이 있다. 도 6에서, 바닥의 그레이 노드는 별개의 레벨을 형성하지 않는다. 쿼리 과정이 제 4레벨에서 종료할 때, 어느 룰이 매칭되는지 나타낸다.
- <60> 2. 트리에서 각각의 노드는 부모 노드의 룰세트의 서브세트인 룰세트를 포함한다. 트리의 루트노드는 분류기에서 모든 룰을 포함하도록 정의된다.
- <61> 3. i번째 레벨의 노드 a(루트노드는 제 1레벨에 있는 것으로 정의된다)는 노드 a에 포함된 모든 룰의  $F_i$  필드의 값에 의거하여 i+1번째 레벨에서 그 자식노드를 생성한다.
- <62> 필드  $F_i$ 의 특징에 의거하여 자식노드 생성에 대한 2개의 다른 과정이 있다.
- <63> (a)  $F_i$ 가 프리픽스로 특정화되면, a의 자식노드의 수는 a의 룰세트에서  $F_i$ 의 다른 값의 수와 같게 된다. 그래서, 각각의 자식노드는 다른 프리픽스와 결합된다. 자식노드(b)가 프리픽스(p)와 결합되어 있다고 가정하면, b의 룰세트에 포함되어 있는 룰(r)(r은 또한 a에도 포함되어 있다)의  $F_i$ 값이 같거나 p의 프리픽스가 된다.
- <64> 예를 들면, 도 6에서 루트노드는 모든 7개 룰을 포함하고, F1 필드에는 4개의 다른 프리픽스 \*, 0\* , 00\* , 10\* 가 있어서, 4개의 자식 노드가 생성된다.
- <65> 프리픽스( 0\* )와 결합되어 있는 노드(x)는 4개의 룰(R1~R7)을 포함한다. R4~R6의 F1값은 0\* 이고, 이것은 프리픽스와 결합된 것이다. R7의 F1값은 \* 이고, 이것은 0\*의 프리픽스이다.

- <66> (b)  $F_i$ 가 레인지에서 특정화되면, 레인지를 넘버라인에 투사하고, 한 세트의 구간을 얻는다. 각각의 구간( $I$ ), 자식노드( $b$ )가 생성된다. 룰( $r$ )은  $b$ 의 룰세트에 포함되고,  $r$ 의  $F_i$ 필드에 의해 특정화된 범위는 1을 포함한다. 예를 들면, 노드( $y$ )는 구간  $[10, 10]$ 을 갖고 단일 포인트인 노드( $y'$ ), 구간 $[6,6]$ 을 갖는 노드( $y''$ ) 및 구간 $[4,5]$ 와  $[6,7]$ 을 갖는 노드( $y'''$ )의 3개의 자식노드를 생성한다.
- <67>  $i$ 번째 레벨에서 노드( $a$ )의 룰세트는 동일 레벨에 있는 모든 노드의 룰세트 중에서 독특하다.
- <68>  $i-1$ 번째 레벨에서 2개의 노드( $b, c$ )가 공통의 자식노드( $a$ )를 가지면, 단지 한개의 노드, 노드( $a$ )가 생성되고, 그것을 공유한다. 노드가 다중 포인터에 의해 지적될 때 노드 공유가 발생함을 도 6을 통해서 알 수 있다. 노드 공유는 그리드-오브-트리에서 스위칭 포인터의 메카니즘과 유사하지만, 그리드-오브-트리가 비트레벨에서 노드 공유를 얻는 반면, 본 발명에서 제안된 방식은 필드레벨에서 노드 공유를 얻는다. 필드레벨 공유는 룰 복사를 피할 수 없기 때문에 쿼리 성능을 개선하지만 저장 요구는 증가한다. 도 6에 나타낸 것같이, 룰( $R_7$ )은 제 2레벨에서 4개의노드에 저장된다.
- <69> 다음은 필드 레벨 트리의 노드 구조와 쿼리 과정에 대하여 설명한다.
- <70> 각각의 필드는 프리픽스 또는 레인지 폼으로 통상 구체화되고, 각각의 구체화는 그 자신이 선호하는 데이터 구조와 검색 알고리즘을 갖는다. 그러므로, 분류기는 2그룹의 필드를 갖는다. 제 1그룹은 프리픽스 구체화이고 제 2그룹은 레인지 구체화이다.

- <71> 표 2에 나타난 분류기에 있어서, F1과 F2는 제 1그룹이고, F3와 F4는 제 2그룹이다. 필드레벨트리는 제 1그룹의 필드가 상위레벨에 나타나고 제 2그룹의 필드는 하위레벨에 나타나는 구조로 조직화된다.
- <72> 단지 프리픽스 형태의 필드를 포함하는 제 1그룹에 대하여, TCAM은 프리픽스를 저장하고 그들을 검색하기 위해 사용된다. 더욱이, TCAM은 동시에 다중 필드를 수용할 수 있기때문에, 필드의 제 1그룹의 쿼리는 단지 한번의 메모리 접속에 의해 실행될 수 있다.
- <73> 도 7은 도 6의 프리픽스 록업을 위한 데이터구조(즉, 레벨1, 2)를 도 5의 분류엔진 내에 TCAM(Ternary Content Addressable Memory)을 사용해서 단축한 구성도이다. 도 7에서, F1, F2 필드에 대하여 단지 하나의 레벨만이 존재한다. 루트노드는 도 5에서 제 3레벨에 최초로 있는 7개의 자식노드를 갖는다. 각각의 제 2레벨노드는 서로 결합된 F1/F2 프리픽스쌍을 갖는다. 이러한 각각의 프리픽스쌍은 TCAM에서 엔트리의 콘텐츠이다. 프리픽스쌍은 도 6에서 트리 구조로부터 유도된다. 도 7의 제 2레벨에서의 노드에 대응하여, 도 6의 제 3레벨에서 각각의 노드(x)에 대하여, 루트노드로부터 노드(x)까지 가장 작은 프리픽스 길이의 합으로경로를 발견한다. 이 경로에 따른 프리픽스는 도 7에서 노드의 프리픽스쌍을 형성한다. 모든 프리픽스쌍은 TCAM에서 프리픽스 길이가 감소하는 순서로 배열된다. 동일한 길이를 갖는 프리픽스쌍에 대하여, 그들의 상관순서는 임의이다. 도 7에서 트리구조의 일예에 대한 TCAM의 콘텐츠는 표 3에 나타나 있다. 제 2레벨에서 적당한 노드는 전체의 쿼리과정을 계속한다고 결정된다.
- <74> FLT와 TCAM의 특징으로 조합하여 단계가 축소되며 표 2의 프리픽스 록업부분을 TCAM내에 구현한 예가 표 3이다.



## &lt;75&gt; 【표 3】

엔트리	프리픽스쌍	노드명	길이의 합
1	00*/110*	f	5
2	00*/11*	e	4
3	10*/1*	g	3
4	0*/01*	d	3
5	0*/10*	c	3
6	0*/1*	b	2
7	*/00*	a	2

<76> 프리픽스쌍을 감소하는 순서로 TCAM에서 배열함으로써, TCAM에서의 검색결과가 정확한 것이 보증되고, 이것은 제 2레벨에서 정확한 노드는 전체의 쿼리과정을 계속하기 위해 결정된다.

<77> 도 8a 및 도 8b는 도 5의 분류엔진내의 레인지룩업을 위한 마이크로 엔진에서 수행되는 k-way search를 설명하기 위한 원리도이다. 2차분류의 가장 큰 성능을 좌우하는 k-way search의 k값을 메모리 인터페이스의 크기에 따라서 정해진다. 주어진 환경하에서 k값은 클수록 좋다.

<78> 분류되어야할 패킷헤더가 주어지면, 제 1그룹에 속하는 필드가 추출되고, 검색을 위해 TCAM에 주어진다. TCAM으로부터의 출력은 다음에 접속되어야할 제 2레벨에서 노드를 표시한다. TCAM은 모든 프리픽스형태 필드를 수용하기때문에, 쿼리 프로세스의 나머지는 레인지 구체화 필드에 의지한다. 제 2 또는 하위레벨에 존재하는 노드에 대하여, 각각의 노드에 대하여 바이너리 검색 트리(또는 외부 메모리 대역폭에 따라서 k-way 검색 트리)를 사용한다. 예를 들면, 트리의  $i(i>1)$ 번째 레벨에서 노드(a)가 있다. a의 룰세트에서 룰의  $i$ 번째 필드를 넘버라인에 프로젝트한 후, 도 8a에 도시된 것같이 8개의

중점( $E_1 \sim E_8$ )을 갖는 7개 인버벌( $I_1 \sim I_7$ )이 얻어진다. 이들 인터벌을 조직화하기 위해 3-way 검색트리를 이용하면, 도 8b에 나타낸 것같은 트리구조가 얻어진다. 이것은 4개의 블록을 갖는 2레이어 트리이다. 각각의 블록은  $k$ -포인터와  $k-1$  중점을 포함한다. 내부 블록에서 포인터는  $k$ -way 검색트리에서 다른 블록을 지시하는 반면에, 리프(leaf) 블록에서 포인터는 필드레벨트리에서 다음 레벨 노드를 지시한다.  $k$ -way 검색트리에서 검색과정을 예를 들어 설명한다. 인터벌( $I_3$ )내에 포인트( $P$ )가 존재한다고 가정하면, 검색과정은 루트블록( $x$ )로부터 시작한다.  $P$ 를  $x$ 에 저장된 2개의 중점( $E_3, E_6$ )와 비교함으로써, 그들 사이의 순서가  $E_3 < P < E_6$ 인 것을 알 수 있다. 인터벌( $I_3$ )와 결합된 제 1포인터가 필드레벨트리의 다음 레벨에서의 노드의 뒤에 와야 한다.

<79>  $k$ -way 검색은 레인지 룩업 문제를 위한 효과적인 알고리즘이다.  $k$ -way 검색 트리의 레이어의 수는  $\log_k M$ 에 의해 결정될 수 있고,  $M$ 은 인터벌의 수이다.  $k$ -way 검색 트리에서 각각의 블록은 메모리에 저장된 기본 단위이고, 한번의 리드/라이트 동작에 대하여 한번의 메모리 접속을 필요로 한다. 그래서, 검색과정동안, 메모리 접속의 수는  $k$ -way 검색트리의 레이어의 수와 동일하다. 여기서  $k$ 의 수는 블록 크기에 의해 제한되고 블록크기는 메모리 대역폭에 의해 결정된다.

<80> 필드 레벨 트리의 쿼리 과정은 모든 필드에 대하여 프리픽스 구체화를 갖는 TCAM으로부터 시작한다. 레인지 구체화를 갖는 필드에 도달한 후, 쿼리 과정은 한번에 하나의 레벨에 진행하고,  $k$ -way 검색은 접속할 다음의 자식노드를 찾기 위해 실행된다. 쿼리 과정은 리프 노드가 도달한 때 종료되고, 정합된 룰이 결과로서 복귀된다.

【발명의 효과】

<81>        본 발명에 의하면, 필드단위의 트리를 전개함으로써 뛰어난 쿼리 성능이 확보된 초고속 네트워킹을 위한 패킷 분류를 구현할 수 있다. 또한, 약 50만개의 분류기 룰을 처리할 수 있다.

**【특허청구범위】****【청구항 1】**

분류를 위해 멀티필드를 갖는 패킷을 계층적 구조로 필드별로 조직화한 필드레벨트리를 생성하여 유지하는 주처리부; 및

쿼리와 갱신업무를 수행하며, IP 소스/목적지 어드레스 룩업으로 대표되는 프리픽스 룩업을 처리하는 1차 분류부와, 상기 1차 분류부의 결과에 의거하여 그에 속한 레인지 룩업범위를 처리하기 위해 해당 필드별로 분류를 진행하는 2차 분류부를 구비하는 분류엔진;을 포함하는 것을 특징으로 하는 필드레벨트리를 이용한 패킷분류장치.

**【청구항 2】**

제 1항에 있어서,

상기 분류엔진은 분류 프로세서와 메모리를 포함하는 것을 특징으로 하는 필드레벨트리를 이용한 패킷분류장치.

**【청구항 3】**

제 1항에 있어서,

상기 주처리부와 상기 분류엔진은 브로드캐스팅 버스로 연결되어 있는 것을 특징으로 하는 필드레벨트리를 이용한 패킷분류장치.

**【청구항 4】**

제 1항에 있어서,

상기 1차 분류부는 프리픽스 형태의 필드를 저장하고 그것을 검색하는 TCAM을 사용하는 것을 특징으로 하는 필드레벨트리를 이용한 패킷분류장치.

**【청구항 5】**

제 1항에 있어서,

상기 2차 분류부는 용도 및 스펙에 따라 적당한 k값을 갖는 k-way search방법을 사용하는 것을 특징으로 하는 필드레벨트리를 이용한 패킷분류장치.

**【청구항 6】**

제 1항에 있어서,

상기 주처리부는 상기 브로드 캐스팅 버스를 통해 상기 복수의 분류엔진에 갱신지시를 보내고, 상기 갱신지시를 받은 분류엔진은 상기 메모리의 콘텐츠를 변경하도록 지시하는 것을 특징으로 하는 필드레벨트리를 이용한 패킷분류장치.

**【청구항 7】**

제 1항에 있어서,

상기 필드레벨트리는 제 1그룹의 필드가 상위레벨에 나타나고 제 2그룹의 필드는 하위레벨에 나타나는 구조로 조직화되는 것을 특징으로 하는 필드레벨트리를 이용한 패킷분류장치.

**【청구항 8】**

제 1항에 있어서,

상기 필드레벨트리는 어느 레벨에서 2개의 노드가 공통의 자식노드를 가지면, 단 하나의 노드가 생성되고 그것을 공유하는 것을 특징으로 하는 필드레벨트리를 이용한 패킷분류장치.

**【청구항 9】**

제 1항에 있어서,

상기 필드레벨트리는 프리픽스 록업을 위한 레벨은 서로 결합된 프리픽스쌍을 갖는 단지 하나의 레벨로서 존재하는 것을 특징으로 하는 필드레벨트리를 이용한 패킷분류장치.

**【청구항 10】**

라우팅 시스템의 패킷 분류방법에 있어서,

주어진 멀티필드를 갖는 패킷을 필드별로 전개하여 필드레벨트리를 형성하는 제 1 단계;

상기 필드레벨트리를 이용하여 패킷 분류 규칙들에 대해 프리픽스 록업을 행하는 제 2단계; 및

프리픽스 록업이 행해진 이후, 레인지 록업을 행하는 제 3단계;를 포함하는 필드레벨트리를 이용한 패킷분류방법.

**【청구항 11】**

제 10항에 있어서,

상기 필드레벨트리는 제 1그룹의 필드가 상위레벨에 나타나고 제 2그룹의 필드는 하위레벨에 나타나는 구조로 조직화되는 것을 특징으로 하는 필드레벨트리를 이용한 패킷분류방법.

**【청구항 12】**

제 10항에 있어서,

상기 필드레벨트리는 어느 레벨에서 2개의 노드가 공통의 자식노드를 가지면, 단지  
한개의 노드가 생성되고 그것을 공유하는 것을 특징으로 하는 필드레벨트리를 이용한  
패킷분류방법.

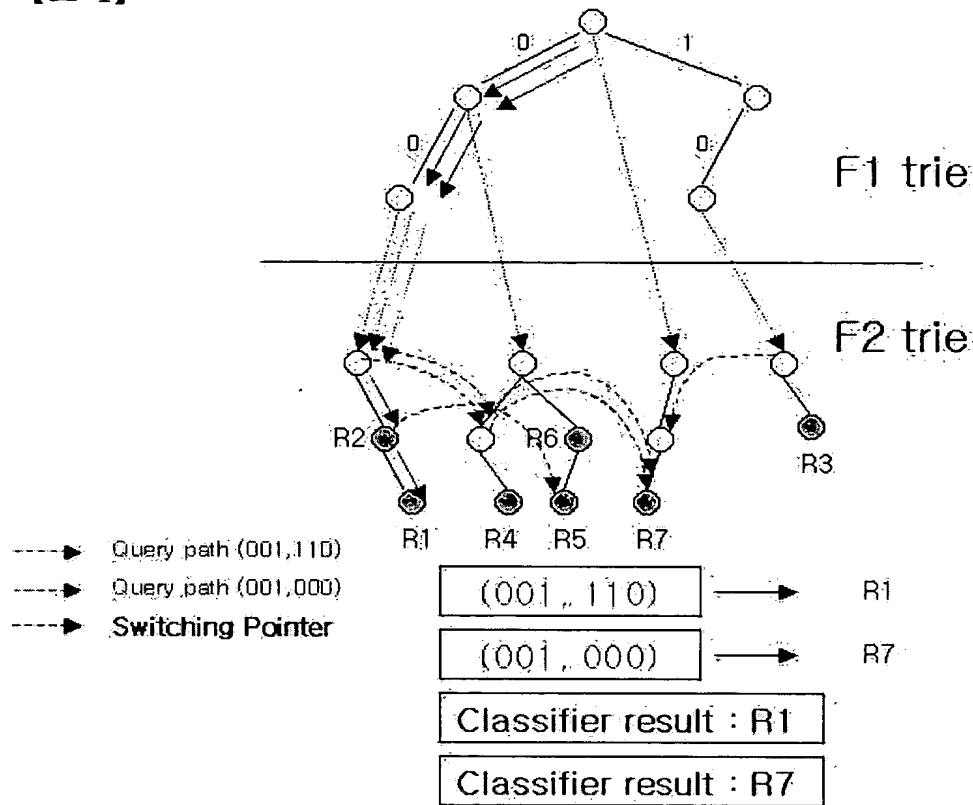
**【청구항 13】**

제 10항에 있어서,

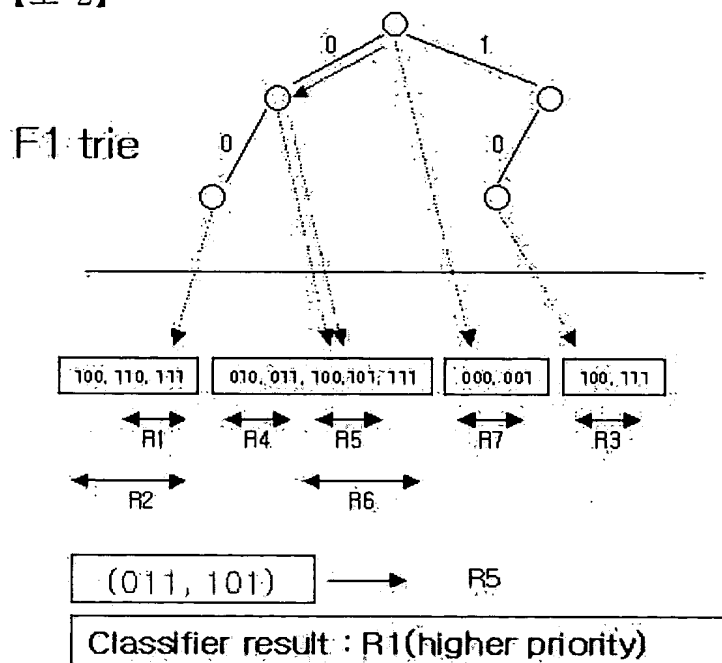
상기 필드레벨트리는 프리픽스 룩업을 위한 레벨이 서로 결합된 프리픽스쌍을 갖는 단지  
하나의 레벨로서 존재하는 것을 특징으로 하는 필드레벨트리를 이용한 패킷분류방법.

【도면】

【도 1】

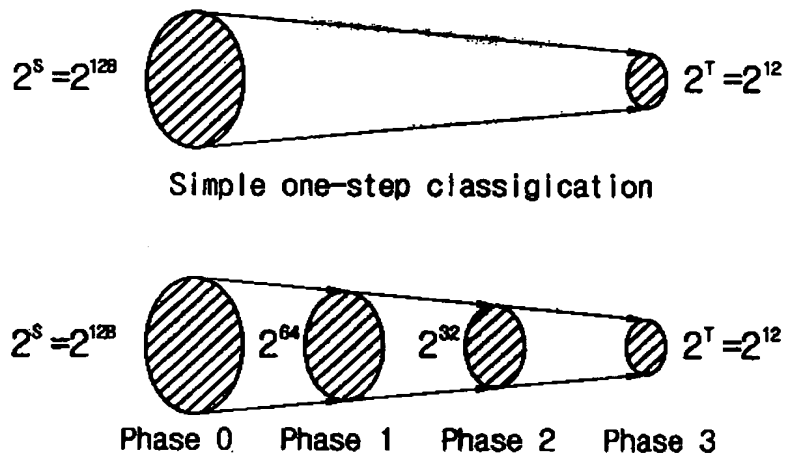


【도 2】

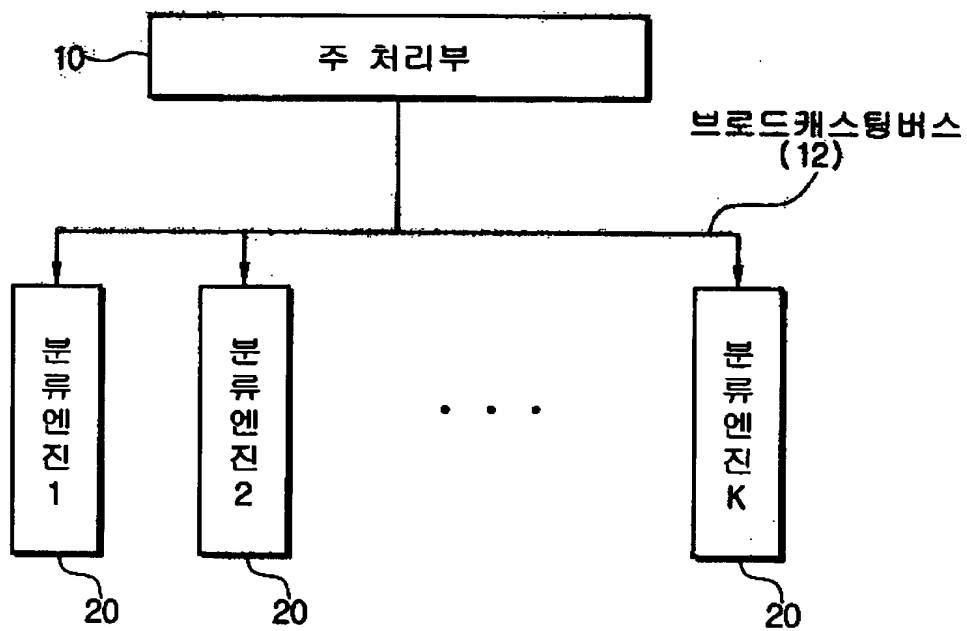




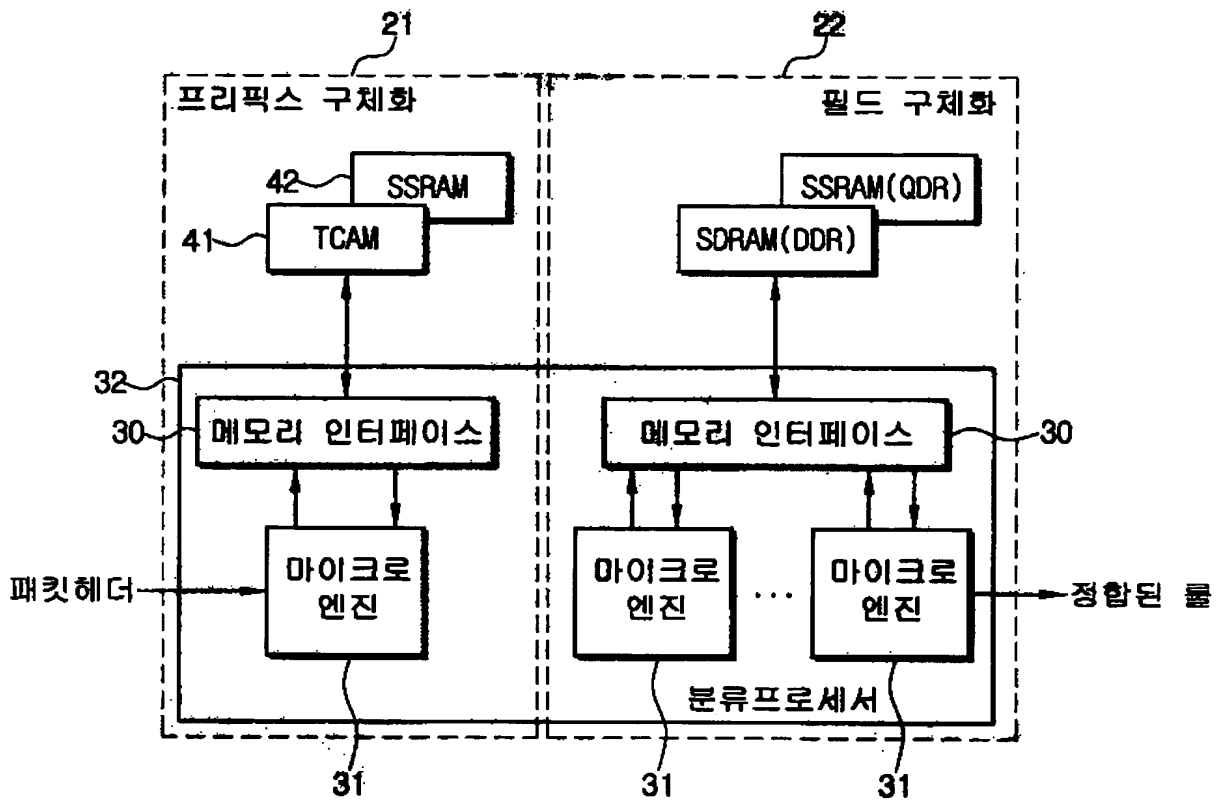
【도 3】



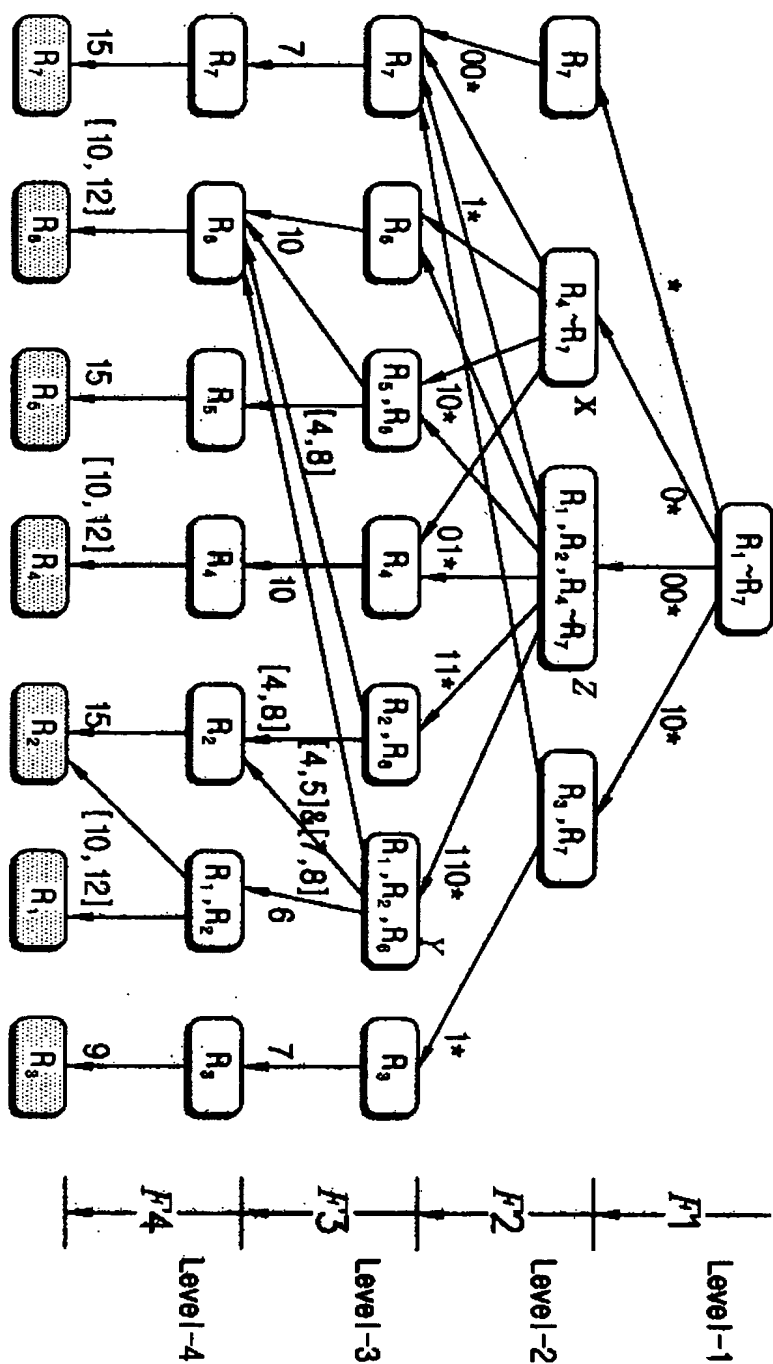
【도 4】



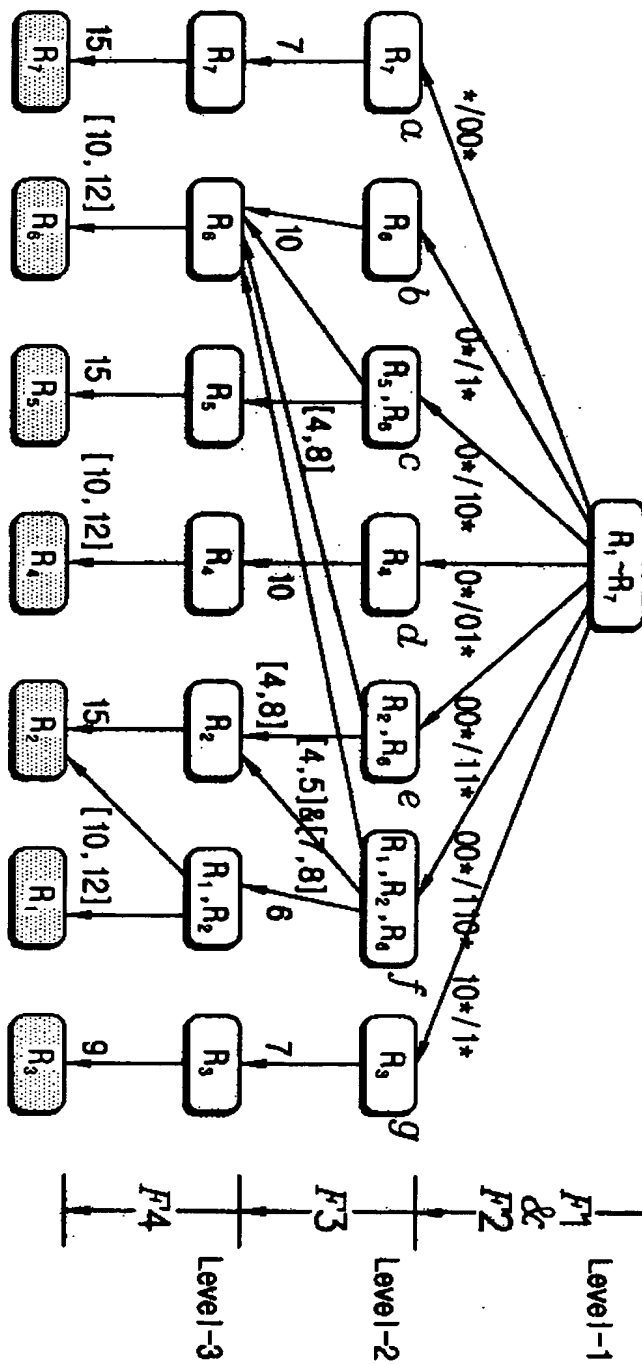
【도 5】



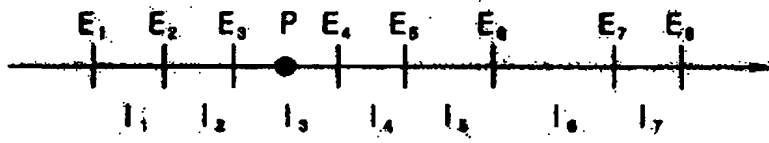
【도 6】



【도 7】



【도 8a】



【도 8b】

